

Remarks

Claims 1, 3-10, 12-13, and 15-20 are pending in this application. Independent claims 1 and 8 have been amended. The Examiner has rejected claims 1, 3-10, 12-13, and 15-20 as being obvious under 35 U.S.C. 103(a) over “Kernel Korner Writing a Linux Driver” by Matia (hereinafter “Matia”) in view of “SCONE: Using Concurrent Objects for Low-level Operating System Programming” by Itoh (hereinafter “Itoh”), and further in view of U.S. Patent No. 6,754,858 to Broman (hereinafter “Broman”). Applicants note that the correct patent number for Broman is 5,754,858, since U.S. Patent No. 6,754,858 was not issued to Broman.

A. The Combination of Matia, Itoh, and Broman Does Not Establish a Prima Facie Case of Obviousness as to Independent Claims 1, 8, and 13

Applicants respectfully submit that a prima facie case of obviousness has not been established and that a rejection of the pending claims on obviousness grounds is improper. A prima facie case of obviousness requires a showing that all of the claim limitations of the rejected claims are taught or suggested by the prior art. Manual of Patent Examining Procedure 2143 and 2143.03. The establishment of a prima facie case of obviousness requires that *all* the claim limitations be taught or suggested by the prior art. MPEP 2143.01 (emphasis added). “All words of a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (CCPA 1970). Here, a prima facie case of obviousness is not established because (a) the combination of Matia, Itoh, and Broman does not disclose or suggest a device driver having a **pre-compiled module** in executable form; and (b) the combination of Matia, Itoh, and Broman does not disclose or suggest the step of “compiling the service layer against the kernel . . . after each **modification** to the **kernel**.”

1. The Combination of Matia, Itoh, and Broman Does Not Disclose or Suggest a Device Driver having a Pre-compiled Module

The present invention is directed to a system for dynamic device driver support in an open source operating system. Specifically, the present invention addresses the problem of providing a driver having proprietary information concerning the hardware architecture of a manufacturer's computer system in an open source operating system. (Spec., p.9, lines 3-6) An advantage of the present invention is the use of **pre-compiled** driver modules in a device driver, such that the proprietary information of the manufacturer resides in the pre-compiled driver modules and is protected from disclosure in the open source environment. (Spec., p.6, lines 5-10) This way, a user may modify or replace the kernel of an open source operating system and rebuild the device driver (by compiling an open-source service layer against the modified kernel) without disclosing proprietary information. (Spec., p.6, lines 4-10) Because the device drive modules are pre-compiled and are in an executable format, it is difficult to identify the proprietary content within the module. (Spec., p.9, lines 1-5)

In contrast to the present invention, the combination of Matia, Itoh, and Broman fails to teach or suggest a device driver having a **pre-compiled module** in executable form. Matia, at best, discloses that drivers are pieces of software that perform low-level communications with devices and that have internal functions. (Matia, p.2, lines 17-20) Specifically, Matia discusses four types of drivers and the differences in how they communicate with users and devices. (Matia, p.3-4) However, nowhere does Matia teach or suggest that a device driver has a **pre-compiled module** in executable form, wherein a compiled open-source service layer acts as an interface between the kernel of the operating system and the pre-compiled module. Internal driver functions are not the same as pre-compiled modules, and Matia does not discuss the pre-compilation of any modules or driver sections. Itoh does not

remedy the deficiencies of Matia, as Itoh also does not discuss pre-compiled modules of device drivers. Finally, the Examiner has not cited to Broman as disclosing device drivers of any kind, and as such, Broman does not remedy the deficiencies of Matia and Itoh. All of the claim limitations are not taught or suggested by the prior art. For at least this reason, the Examiner's obviousness rejection of claims 1, 8, and 13 should be withdrawn.

2. The Combination of Matia, Itoh, and Broman Does Not Disclose or Suggest Compiling the Driver against the Kernel after Each Modification to the Kernel

The Examiner states in the Response to Arguments that Matia teaches that a driver is the part of the OS that manages communications with devices and that the OS is composed of a set of drivers, which are pieces of software that perform the low-level communication with each device. (Office Action, p.7) The Examiner concludes from these statements that it is clear that when a driver is compiled *after modification to the **driver***, it means that the driver is recompiled against the kernel and is integrated into the kernel. (Office Action, p.7) Applicants disagree with the Examiner's assertions. First, Applicants note that the Examiner has not addressed the Applicants' previous argument that Matia fails to disclose a driver recompilation against the kernel after each modification to the **kernel**. Matia teaches only that a driver is recompiled after modification to the **driver itself**. The Examiner has only referred to a situation in which a driver is compiled after modification to the **driver itself**, which is not cited in the claims of the present invention.

As Applicants have stated before, Matia concerns the recompilation of the device driver following a modification to the **device driver**. As an example, on page 8 of Matia, under the heading "Implementation of Driver Functions", the user is given instructions on "programming your own driver." These instructions continue through page 10 and concern steps for recompiling the driver following a modification *to the driver itself* and **not** to the kernel, as

required by the claims of the present invention. Page 10 of Matia, for example, concerns the “task of integrating the driver into the kernel” and describes the step of “re-compile the driver.” The recompilation of Matia, however, occurs after a modification to the **driver**, and **not** after each modification to the **kernel**, as required by the independent claims of the application.

Additionally, as Applicants have stated before, on page 11 of Matia, the user is told that it is recommended that the driver be compiled **alone** before linking the kernel. This is not the same as compiling the service layer **against the kernel**, as required by the independent claims. Additionally, Matia, on page 11, describes configuring the kernel after compiling the driver alone. Matia does not disclose the element of the independent claims that requires that the server layer be compiled against the kernel **after** each modification to the **kernel**. The Examiner does not refer to Itoh or Broman as disclosing or suggesting these elements. Because these elements of the claims are not taught or suggested by Matia in combination with Itoh and Broman, a prima facie case of obviousness cannot be established by the combination of Matia, Itoh, and Broman. As such, the rejection of claims 1, 8, and 13 should be withdrawn.

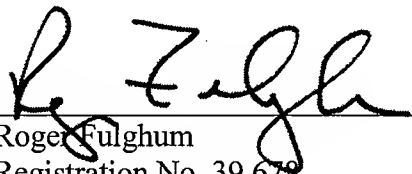
B. Dependent Claims 3-7, 9-10, 12, and 15-20

Dependent claims 3-7, 9-10, 12, and 15-20 will not be discussed individually herein, as these claims depend, either directly or indirectly, from an otherwise allowable base claim.

Conclusion

Applicants respectfully submit that the pending claims 1, 3-10, 12-13, and 15-20 of the present invention, as previously amended, are allowable. Applicants respectfully request that the rejection of the pending claims be withdrawn and that these claims be passed to issuance.

Respectfully submitted,



Roger Fulghum
Registration No. 39,678

Baker Botts L.L.P.
910 Louisiana
One Shell Plaza
Houston, Texas 77002-4995
(713) 229-1707

Baker Botts Docket Number: 016295.0732

Date: January 7, 2008